Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Vision and Image Understanding 99 (2005) 303–331

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Visual tracking and recognition using probabilistic appearance manifolds

Kuang-Chih Lee [a,*], Jeffrey Ho [b], Ming-Hsuan Yang [c], David Kriegman [b]

[a] *Beckman Institute and Computer Science Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

[b] *Computer Science and Engineering Department, University of California at San Diego, La Jolla, CA 92093-0114, USA*

[c] *Honda Research Institute, 800 California Street, Mountain View, CA 94041, USA*

Received 13 August 2004; accepted 28 February 2005
Available online 12 April 2005

Abstract

This paper presents an algorithm for modeling, tracking, and recognizing human faces in video sequences within one integrated framework. Conventional video-based face recognition systems have usually been embodied with two independent components: the tracking and recognition modules. In contrast, our algorithm emphasizes an algorithmic architecture that tightly couples these two components within a single framework. This is accomplished through a novel appearance model which is utilized simultaneously by both modules, even with their disparate requirements and functions. The complex nonlinear appearance manifold of each registered person is partitioned into a collection of submanifolds where each models the face appearances of the person in nearby poses. The submanifold is approximated by a low-dimensional linear subspace computed by principal component analysis using images sampled from training video sequences. The connectivity between the submanifolds is modeled as transition probabilities between pairs of submanifolds, and these are learned directly from training video sequences. The integrated task of tracking and recognition is formulated as a maximum a posteriori estimation problem. Within our framework, the tracking and recognition modules are complementary to each other, and the

[*] Corresponding author.
E-mail address: klee10@uiuc.edu (K.-C. Lee).

doi:10.1016/j.cviu.2005.02.002

capability and performance of one are enhanced by the other. Our approach contrasts sharply with more rigid conventional approaches in which these two modules work independently and in sequence. We report on a number of experiments and results that demonstrate the robustness, effectiveness, and stability of our algorithm.

Keywords: Face tracking; Face recognition; PCA; Appearance manifold

1. Introduction

In the past few decades, there has been intensive research and great strides in designing and developing algorithms for face recognition with still images. Only until recently has the problem of face recognition with video sequences started to attract the attention of the research community [1–4]. This can be partly attributed to the recent advance in computer hardware. In particular, with low cost cameras and sufficiently powerful personal computers, it is now possible to inexpensively implement a real-time face tracking system (e.g. [5,6]) with good performance. This capability is the prerequisite for developing real-time video face recognition applications.

Compared with conventional still image face recognition, video face recognition offers several challenges and opportunity. First, there is the "alignment" problem between the tracking and the recognition modules. A video-based face recognition system invariably has two components, i.e., tracking and recognition modules. Since tracking and recognition problems have been studied intensively but separately in the past, these two modules are usually implemented independently and work in sequence. Without any alignment between the two modules, the images returned by the tracker generally are not be in good agreement with the appearance model used by the recognition module,[1] i.e., misaligned images. Unfortunately, virtually all appearance-based recognition techniques are sensitive to misalignments. Therefore, some mechanism should be in place to ensure that the images returned by the tracking module can be correctly processed by the recognition module.

Second, there is the problem of modeling appearance variation of faces for both the tracking and recognition modules. At the heart of any tracking or recognition algorithm is an internal representation which defines the allowable variation in appearances of the object to be tracked or recognized. Factors such as changes of viewpoint, shape (deformations, articulations), and illumination, individually or combined, can cause significant image variations in a dynamic environment. (See Fig. 1). For appearance-based methods, some (if not all) of these image variations should be modeled in order to produce robust results. However, due to their different missions, tracking and recognition modules generally place different emphasis and requirement on their internal model or representation. For

[1] In this paper, our main focus is on appearance (or image)-based recognition methods. For face recognition, it has been argued [7,8] that feature-based techniques are generally less stable and accurate.

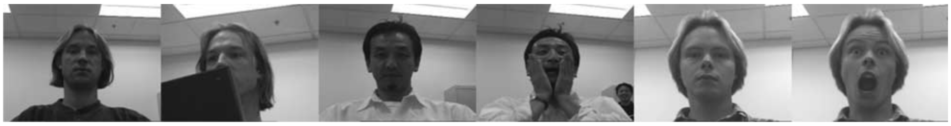


Fig. 1. Other important image variations for video-based face recognition include occlusion by an external object, expression variation and the combination of the two. For face recognition methods using still images, identification is generally quite difficult with these images.

recognition, the model is required to accurately capture subtle differences between the appearances of different enrolled individuals in order to correctly recognize them. For tracking, such fine granularity in detail is unnecessary. Instead, a model that captures common image features of human faces is preferred, and perhaps more importantly, the model should be simple and efficient so that the tracking module can complete its task quickly. Therefore, in developing a combined tracking and recognition system, our challenge is to design a model that is both accurate and efficient.

Finally, unlike still-image recognition, video-based recognition provides the opportunity to correctly identify individuals in frames, even though there is not strong support for a decision solely with image content. As illustrated by the frames in Fig. 1 in which the faces are occluded, have widely varying facial expression, or have both events occurring, most still-image systems would likely make mistakes. One expects that, in a real-world situation, a video sequence will be punctuated by episodes similar to those shown in the figure. Yet, the recognition decision prior to these episodic circumstances can be utilized to assist in determining the correct identification. What is needed, therefore, is a principled method for integrating information and decisions from earlier frames so that robust and stable recognition results are still possible when the conditions are more difficult.

In this paper, we propose a unified framework and appearance model that address the aforementioned three problems. Based on this framework, we propose an algorithm that can simultaneously track and recognize human faces. Our solution to the alignment problem is to abandon the typical two-component architecture in favor of a tightly integrated tracking/recognition algorithm, in which both the tracker and recognizer share the same appearance model. The sharing of the model increases the likelihood of tracker returning results that are in good alignment for the recognizer. Furthermore, while the recognition module keeps a detailed appearance model for each registered individual, at each frame, the tracker only uses a portion of the appearance model of an individual identified by the recognizer. Therefore, the actual appearance model used by the tracker is small but accurate, and it lessens the tracker's computational load considerably.

The appearance model we propose is based on the concept of the appearance manifold [9], and the actual model is a piecewise linear approximation of the appearance manifold, i.e., a collection of affine subspaces in the image space. During training, we apply a clustering algorithm to partition the training images into clusters, and the images in each cluster usually come from neighboring poses. Principal component analysis (PCA [10]) can be applied to images in each cluster to yield a low-dimensional linear subspace approximation. The connectivity among the linear subspaces is represented by a transition matrix that encodes the likelihood of observing transitions between a pair of subspaces in a video sequence. Finally, recognition results from the previous frames and information from the current frame are considered in a Bayesian framework, and a maximum likelihood estimate yields the recognition result for the current frame. For the experiments, we have collected more than 50 video sequences with varying degrees of difficulty, and scores of experiments were performed to validate each aspect of our algorithm. The experiments demonstrate that our approach does indeed improve both the tracking and recognition performance, especially when compared with algorithms based on the existing techniques.

The rest of this paper is organized as follows. We briefly summarize the most relevant works in Section 2. In Section 3, we detail our probabilistic appearance model and the online tracking and recognition algorithm. Tracking and recognition experiments on a large and difficult collection of video sequences are reported in Section 4. We conclude this paper with remarks and future work in Section 5. An early version of our approach to video face recognition, but with a different tracking algorithm, was presented in [11].

2. Related work

While numerous tracking and recognition algorithms have been proposed, in the vision community, these two topics were usually studied separately. For human face tracking, many different techniques have been proposed, such as subspace-based methods [6,12], pixel-based tracking algorithms [13], contour-based tracking algorithms [5,14,15], and global statistics of color histograms [5,16]. Likewise, there is a rich literature on face recognition published in the last 15 years (see [17–19] for surveys). However, most of these works deal exclusively with still images, and in several cases, [20–22], algorithms for still images are generalized in a straightforward way to perform video face recognition. In these algorithms, the still image recognition algorithm is applied independently for each frame and temporal voting is used to improve the identification rate. Among the few attempts aiming to address the problem of video-based face recognition in a more systematic and unified manner, the methods by Zhou and Chellappa [23], Krueger and Zhou [2], and Liu and Chen [4] are the most relevant.

Zhou and Chellappa [23] proposed a generic framework to track and recognize human faces simultaneously by adding an identity variable to the state vector in a sequential importance sampling method. They then marginalized over all state vectors to yield an estimate of the posterior probability of the identity variable. Though this probabilistic approach aims to integrate motion and identity information over time, it nevertheless considers only identity consistency in the temporal domain and thus may not work well when the target is partially occluded. Furthermore, it is not clear how one can extend this work to deal with large 3-D pose variation.

Krueger and Zhou [2] applied an on-line version of radial basis functions to select representative face images as exemplars from training videos, and in turn this facilitates tracking and recognition tasks. The state vector in this method consists of affine parameters as well as an identity variable, and the state transition probability is learned from affine transformations of exemplars within training videos in a way similar to [24]. Since only 2-D affine transformations are considered, this model is effective in capturing small 2-D motion but may not work well with large 3-D pose variation or occlusion.

Though no new tracking algorithm was presented, Liu and Chen [4] proposed a video face recognition algorithm based on a hidden Markov model (HMM). At a high-level, their recognition algorithm closely resembles ours in several aspects since our framework also admits a Markovian interpretation. However, our algorithm admits a clear and concise geometric interpretation in terms of the appearance manifolds in the image space, while their algorithm focuses on a more probabilistic framework. The advantage of having a concise geometric interpretation is that many aspects of our algorithm can be made transparent, both conceptually and implementation-wise. Li et al. [25] applied piecewise linear models to capture local motion and a transition matrix among these local models to describe nonlinear global dynamics. They applied the learned local linear models and their dynamic transitions to synthesize new motion video such as choreography. Our work bears some resemblance to their method in the sense that both methods utilize local linear models, something advocated in several prior works [9,26,27], and both learn the relationships among these models [28–31]. In our work, the dynamics is incorporated in a larger probabilistic framework in which the likelihood of the local linear models are propagated through the transition matrix (i.e., utilizing temporal information) with the aim of producing stable and robust face recognition results.

Although similarities exist between our algorithm and the works just cited above, our work differs from theirs in two important aspects. First, none of these works emphasize the importance of combining tracking and recognition into one tightly coupled system. In fact, these works (except [25]) are all recognition algorithms, and the important issue of how to provide well-aligned images for the recognition algorithm under difficult imaging conditions seems to have been neglected. Our point has been that it is difficult to consistently provide good quality tracking result when a person is undergoing significant pose changes (and other tricky factors). Consequently, an important problem for us is how to organize and unify the two seemingly disparate processes, tracking and recognition, into one single algorithmic framework so that each can improve and enhance the performance of the other. Perhaps because of the lack of a robust and stable tracker, the experimental results reported in these prior works seem to have focused on test videos with a limited range of views (e.g., close to frontal). This limited pose variation can be sufficiently modeled by just one linear subspace, thereby making the tracking and recognition problem easier. Furthermore, the importance of modeling and utilizing dynamics as well as pose transitions cannot be fully revealed from such test videos.

Finally, we remark that there are several algorithms that aim to extract 2-D or 3-D face structure from video sequences for recognition and animation [1,8,32–38]. However these methods require meticulous and complicated procedures to build 2-D or 3-D models, and they do not fully exploit temporal information for recognition.

3. Mathematical framework

3.1. Motivations

It has been shown that the set of images of an object under all viewing conditions can be considered as a low-dimensional manifold in the image space [9]. For video face recognition, the foremost important image variation that needs to be adequately modeled is due to pose variation, the relative orientation between the camera and the object, and we limit ourselves to this. Other important image variations such as shape changes (e.g., expression variation) and partial occlusions are not directly modeled in this work. Although such variations are likely to occur in video sequences, we will consider their occurrences to be episodic, and tracking and recognition under these episodic circumstances will be tackled in our framework with the aid of a probabilistic method detailed later.

If the appearance manifold of a face is known, tracking and recognition become straightforward. Suppose there is a set of N faces (indexed by k) that we wish to track and recognize. Let M_k denote the appearance manifold of person k, and $\{F_1, \ldots, F_l\}$ denotes a video sequence of l frames. For each frame, the tracking/recognition system produces an estimate of the face's location in the image and also its identity. In this work, the location of a face in an image is specified by a rectangular region that contains the face, and the rectangular region is represented by a set $\mathbf{u}$ of five parameters, specifying the rectangular region's center (in image coordinates), its width and height as well as its orientation. If $f(\mathbf{u}, F_t)$ denotes the cropping function (f returns the subimage I of F_t enclosed in the rectangular region specified by $\mathbf{u}$), our tracking and recognition algorithm can be succinctly summarized by the following optimization problem

$$(\mathbf{u}_t^*, k_t^*) = \arg\min_{\mathbf{u},k} d(f(\mathbf{u}, F_t), M_k), \tag{1}$$

where $d(I, M_k)$ denotes the usual L^2 distance between an image and manifold M_k. The pair $(\mathbf{u}_t^*, k_t^*)$ is the tracking/recognition result for frame t.

The simplicity of the form of Eq. (1) disguises its complexity as well as the practical difficulty of trying to solve it directly. First, the domain of the optimization $(\mathbf{u}, k)$, can be very large. Unlike [6], Eq. (1) does not provide a closed form formula for the gradients since this almost always requires M_k to have a closed form description (e.g., algebraic equations), which is not available to us. Therefore, optimization techniques for continuous objective functions are not available. One possible solution is to discretize the domain and solve the optimization problem on the discretized domain, by drawing a large number of samples of $\mathbf{u}$, and finding the minimum among the samples. Note that k indexes a discrete variable not a continuous one; therefore, the actual number of samples is the product of the number of samples for $\mathbf{u}$ and the number N of individuals to be recognized. If the number of samples for $\mathbf{u}$ is large (which is usually the case), even a small N would have generated a great quantity of samples for the algorithm to process and hence invariably limit its performance. To reduce the number of necessary samples, we minimize each variable in Eq. (1) independently, i.e., minimize $\mathbf{u}$ with fixed k and vice versa:

$$\mathbf{u}_t^* = \arg\min_{\mathbf{u}} d(f(\mathbf{u}, F_t), M_{k_{t-1}^*}), \tag{2}$$

$$k_t^* = \arg\min_{k} d(f(\mathbf{u}_t^*, F_t), M_k). \tag{3}$$

The two suboptimization problems correspond exactly to the tracking and recognition problems, respectively. In Eq. (2) we are solving a tracking problem with appearance model provided by M_k, whereas Eq. (3) is a recognition problem using the tracking result $\mathbf{u}$ as the input. Therefore, within this framework, the recognizer uses the tracker's result as input, and it updates the internal appearance model used by the tracker through the identity variable k. The tight coupling between the tracking and recognition components is achieved via the shared appearance models $M_1, \ldots, M_N$. Another difficulty of solving Eq. (1) directly is related to the definition of the L^2 distance $d(I, M_k)$ between an image I and a manifold M_k in the image space. By definition, $d(I, M_k) = d(I, x^*)$ with x^* is a point on M_k having minimal L^2 distance to I (See Fig. 2). Even if an analytic description of M_k were available, finding x^* is generally not an easy problem. In our case M_k is, at best, modeled by a modest number of images sampled from it; therefore, M_k is available to us only through a very coarse and sparse representation with many "gaps" in which we have inadequate or incomplete information. The main focus of our work is to provide an effective definition for $d(I, M_k)$ that works for a coarse representation of M_k.

3.2. Probabilistic face recognition

Probabilistically, we can modify Eq. (3) slightly by defining the conditional probability $p(k|I)$ (given image I, the likelihood that it originated from person k) as

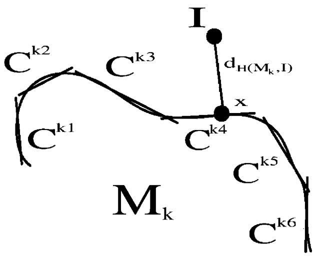


Fig. 2. Appearance manifold. A complex and nonlinear manifold M_k can be approximated as the union of several simpler submanifolds; here, each submanifold C^{ki} is represented by a PCA plane.

$$p(k|I) = \frac{1}{\Lambda} \exp\left(\frac{-1}{\sigma^2} d^2(I, M_k)\right), \tag{4}$$

where Λ is a normalization term, and for a given cropped image I,

$$k^* = \arg\max_k p(k|I). \tag{5}$$

To implement this scheme, one must be able to estimate the projected point $x^* \in M_k$, and then the image to model distance, $d(I, M_k)$, can be computed for a given cropped region I from F_t and for each M_k. However, such distances can be computed accurately only if M_k is known exactly. In our case, M_k is usually unknown and can only be approximated with samples. We provide a probabilistic framework for estimating x^* and $d(x^*, I)$. Note that if we define the conditional probability $p_{M_k}(x|I)$ to be the probability that among points on M_k, x^* has the smallest L^2 distance to I, then

$$d(I, M_k) = \int_{M_k} d(x, I) p_{M_k}(x|I) \mathrm{d}x, \tag{6}$$

and Eq. (3) is equivalent to

$$k^* = \arg\min_k \int_{M_k} d(x, I) p_{M_k}(x|I) \mathrm{d}x. \tag{7}$$

Here, $d(I, M_k)$ can be viewed as the expected distance between an image I and the appearance manifold M_k. If M_k were fully known or well-approximated (e.g., described by some algebraic equations), then $p_{M_k}(x|I)$ could be treated as a δ-function at the set of points with minimal distance to I. When sufficiently many samples are drawn from M_k, the expected distance $d(I, M_k)$ will be a good approximation of the true distance. The reason is that $p_{M_k}(x|I)$ in the integrand of Eq. (6) will approach a delta function with its "energy" concentrated on the set of points with minimal distance to I. In our case, M_k is approximated, at best, through a sparse set of samples, and so we will model $p_{M_k}(x|I)$ with a Gaussian distribution.

Since the appearance manifold M_k is nonlinear, it is reasonable to decompose M_k into a collection of m simpler disjoint submanifolds, $M_k = C^{k1} \cup \cdots \cup C^{km}$, with C^{ki} denoting a submanifold in a decomposition of person k's appearance manifold.

Each C^{ki} is assumed to be amenable to linear approximations by a low-dimensional linear subspace computed through principal component analysis (i.e., a PCA plane). We define the conditional probability $p(C^{ki}|I)$ for $1 \leqslant i \leqslant m$ as the probability that C^{ki} contains a point x with minimal distance to I. With $p_{M_k}(x|I) = \sum_{i=1}^{m} p(C^{ki}|I) p_{C^{ki}}(x|I)$, we have

$$\begin{aligned} d(I, M_k) &= \int_{M_k} d(x, I) p_{M_k}(x|I) \mathrm{d}x = \sum_{i=1}^{m} p(C^{ki}|I) \int_{C^{ki}} d(x, I) p_{C^{ki}}(x|I) \mathrm{d}x \\ &= \sum_{i=1}^{m} p(C^{ki}|I) d(I, C^{ki}). \end{aligned} \tag{8}$$

The above equation shows that the expected distance $d(I, M_k)$ can be treated as the expected distance between I and each C^{ki}. In addition, this equation transforms the integral to a finite summation which is feasible to compute numerically.

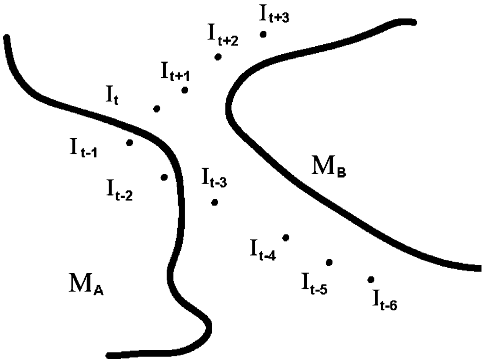


Fig. 3. Difficulty of frame-based tracking/recognition: The two solid curves denote two different appearance manifolds, M_A and M_B. It is difficult to reach a decision on the identity from frame I_{t-3} to frame I_t because these frames have smaller L^2 distance to appearance manifolds M_A than M_B. However, by looking at the sequence of images $I_{t-6} \ldots I_{t+3}$, it is apparent that the sequence has most likely originated from appearance manifold M_B.

For face tracking/recognition in video sequences, we can exploit temporal coherence between consecutive image frames. As shown in Fig. 3, the L^2 distance may occasionally be misleading during tracking/recognition. But if we consider previous frames in an image sequence rather than just one, then the set of closest points x^* will trace a curve on a submanifold C^{ki}. In our framework, this is embodied by the term $p(C^{ki}|I)$ in Eq. (8). In Section 3.3, we will apply Bayesian inference to incorporate temporal information to provide a better estimate of $p(C^{ki}|I)$ and thus $d(I, M_k)$; this will then yield better tracking/recognition performance.

3.3. Computing $p(C_t^{ki}|I_t)$: incorporating dynamics

For tracking/recognition from a video sequence, we need to estimate $p(C_t^{ki}|I_t)$ for each i and k at time t. To incorporate temporal information, $p(C_t^{ki}|I_t)$ should be taken as the joint conditional probability $p(C_t^{ki}|I_t, I_{0:t-1})$ where $I_{0:t-1}$ denotes the frames from the beginning up to time $t-1$. We further assume I_t and $I_{0:t-1}$ are independent given C_t^{ki}, as well as C_t^{ki} and $I_{0:t-1}$ are independent given C_{t-1}^{ki}. Using Bayes' rule and these assumptions, we have the following recursive formulation:

$$\begin{aligned} p(C_t^{ki}|I_t, I_{0:t-1}) &= \alpha p(I_t|C_t^{ki}, I_{0:t-1}) p(C_t^{ki}|I_{0:t-1}) \\ &= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}, I_{0:t-1}) p(C_{t-1}^{kj}|I_{0:t-1}) \\ &= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}) p(C_{t-1}^{kj}|I_{t-1}, I_{0:t-2}) \\ &= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C^{ki}|C^{kj}) p(C_{t-1}^{kj}|I_{t-1}, I_{0:t-2}), \qquad (9) \end{aligned}$$

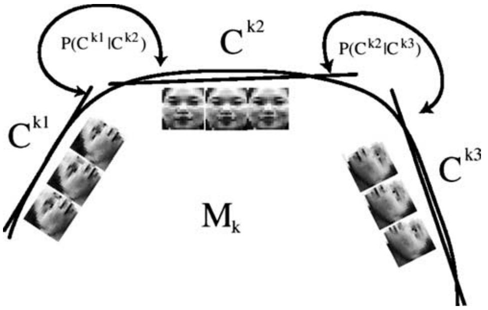


Fig. 4. Dynamics among the submanifolds C^{ki}. The dynamics is learned from training videos which describes the probability of moving from one submanifold C^{ki} to another C^{kj} at any time instant.

where α is a normalization term to ensure a proper probability distribution, and we assume that the transition probability is time invariant.

The temporal dynamics of face motion is captured by the *transition probability* between the manifolds, $p(C_t^{ki}|C_{t-1}^{kj})$. Namely, $p(C_t^{ki}|C_{t-1}^{kj})$ is the probability of the observation I_t being generated from C_t^{ki} given that a previous observation I_{t-1} was generated from the submanifold C_{t-1}^{kj}. The transition probability $p(C_t^{ki}|C_{t-1}^{kj})$ is assumed to be independent to t, and it encodes the temporal coherence of human motion as one cannot move suddenly from C^{ki} to C^{kj} if these two submanifolds are not connected or with low probability (e.g., one cannot move from the leftmost pose to rightmost pose without going through some intermediate pose) (see Fig. 4).

3.4. Learning manifolds and dynamics

For each person k, we collect at least one video sequence containing l consecutive images $S_k = \{I_1, \ldots, I_l\}$. We further assume that each training image is a fair sample drawn from the appearance manifold M_k. There are three steps in the learning algorithm. We first cluster these samples into m disjoint subsets $\{S_1, \ldots, S_m\}$. For each collection S_{ki}, we can consider it as containing points drawn from some submanifold C^{ki} of M_k, and from the images in S_{ki}, we construct a linear approximation to the C^{ki} of the true manifold M_k. After all the C^{ki} have been computed, the transition probabilities $p(C^{ki}|C^{kj})$ for $i \neq j$ are estimated.

In the first step, we apply a K-means clustering algorithm to the set of images in the video sequences. We initialize m seeds by finding m frames from the training videos with the largest L^2 distance to each other. This process can be easily realized by the following greedy search procedure. First an initialized seed is selected randomly, and then the remaining $m - 1$ seeds are iteratively selected to each maximize the average L^2 distance to the seeds already selected. Then the general K-means algorithm is used to assign images to the m clusters. As our goal in performing clustering is to approximate the data set rather than to derive semantically meaningful cluster centers, it is worth noting that the resulting clusters are no worse than twice what the optimal center would be if they could be easily found [39].

Second, for each S_{ki} we obtain a linear approximation of the underlying subset $C^{ki} \subset M_k$ by computing a PCA plane L_{ki} of fixed dimension for the images in S_{ki}. Since the PCA planes approximate the appearance manifold M_i, their dimension is the intrinsic dimension of M, and therefore the dimensions of all PCA planes L_i are taken to be the same.

Finally, the transition probability $p(C^{ki}|C^{kj})$ is defined by counting the *actual* transitions between different S_i observed in the image sequence

$$p(C^{ki}|C^{kj}) = \frac{1}{\Lambda'_{ki}} \sum_{q=2}^{l} \delta(I_{q-1} \in S_{ki})\delta(I_q \in S_{kj}), \tag{10}$$

where $\delta(I_q \in S_{kj}) = 1$ if $I_q \in S_{kj}$, and otherwise it is 0. The normalizing constant Λ'_{ki} ensures that

$$\sum_{j=1}^{m} p(C^{ki}|C^{kj}) = 1, \tag{11}$$

where we set the diagonal terms, $p(C^{ki}|C^{ki})$, to a small constant κ. A graphical representation of a transition matrix with $m = 5$ learned from a training video is depicted in Fig. 5.

With C^{ki} and its linear approximation L_{ki} defined, we can define how $p(I|C^{ki})$ can be calculated. We can compute the L^2 distances $\hat{d}_{ki} = d(I, L_{ki})$ from I to each L_{ki}.

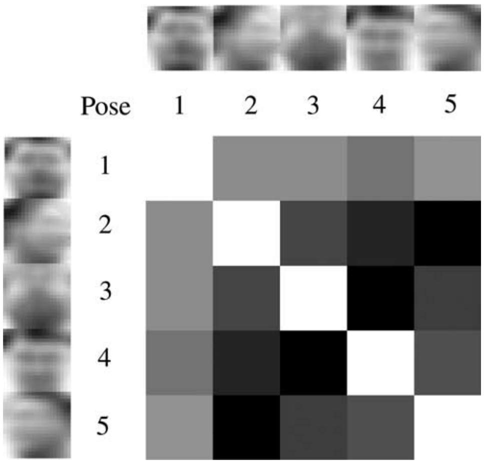


Fig. 5. Graphical representation of a transition matrix learned from a training video. In this illustration, the appearance manifold is approximated by five linear subspaces. The reconstructed center image of each subspace is shown at the top row and column. The transition probability matrix is drawn by the 5×5 block diagram. The brighter block indicates a higher transition probability. It is easy to see that the frontal pose (pose 1) has higher probability to change to other poses; the right pose (pose 2) has almost zero probability to change directly to the left pose (pose 3).

We treat $\hat{d}_{ki}$ as an estimate of the true distance from I to C^{ki}, i.e., $d(I, C^{ki}) = d(I, L_{ki})$. $p(I|C^{ki})$ is defined as

$$p(I|C^{ki}) = \frac{1}{\Lambda''_k} \exp\left(\frac{-1}{2\sigma^2}\hat{d}^2_{ki}\right) \tag{12}$$

with $\Lambda''_k = \sum_{i=1}^{m} \exp(\frac{-1}{2\sigma^2}\hat{d}^2_{ki})$.

3.5. Face tracking and recognition from video sequences

In this section, we outline our tracking/recognition algorithm. The more detailed algorithm is summarized in Fig. 6. Conceptually, the tracker and recognizer compute Eqs. (2 and 3), respectively. Given the current frame F_t from a video sequence and assuming that the tracking result for the previous frame is $\mathbf{u}^*_{t-1}$, the tracker samples a collection of subimages specified by different $\mathbf{u}$ based on a Gaussian distribution centered at $\mathbf{u}^*_{t-1}$.[2] Eq. (13) is evaluated by the tracker (with f as the cropping function)

$$\mathbf{u}^*_t = \arg\min_{\mathbf{u}} d(f(\mathbf{u}, F_t), C^{ki}_{t-1}). \tag{13}$$

The tracker determines a subimage $I_t = f(\mathbf{u}^*_t, F_t)$ which has the shortest distance to the submanifold C^{ki}_{t-1} determined in the previous frame. Next, the recognizer uses the subimage I_t returned by the tracker to compute the distance $d(I_t, M_k)$ for each person k using Eq. (8). Note that $p(C^{ki}_t|I_t)$ has a temporal dependency, and it is computed recursively using Eq. (9). Once all $d(I_t, M_k)$ have been computed, the posterior $p(k|I_t)$ is computed by Eq. (4), and the recognition result is decided by Eq. (7).

4. Experiments and results

In this section, we describe experimental evaluations of our tracking/recognition algorithm. The aim for these experiments is to demonstrate that all of the new ideas introduced in this paper (namely probabilistic modeling of temporal coherence, transition matrix, tracking with identity and local linear approximations) do enhance and improve the performance of the combined tracking/recognition system considerably. Comparisons with algorithms based on well-known existing techniques are also presented.

4.1. Data preparation and training process

Due to the lack of any standard video database for evaluating face tracking/recognition algorithms, we collected a set of 52 video sequences of 20 different persons for the task of testing our system. Each video sequence is recorded by a SONY EVI D30 camera in an indoor environment at 15 frames per second, and each lasted for at

[2] More sophisticated sampling techniques for non-Gaussian distributions (e.g., the CONDENSATION algorithm [40] for incorporating dynamic changes in probability distributions) can also be applied.

Integrated Tracking and Recognition Algorithm:

Input Parameters: $(\Omega,\ S)$

$\Omega = \{\omega_x, \omega_y, \omega_w, \omega_h, \omega_\theta\}$: the set of five parameters for sampling windows on the screen.

S: the number of windows sampled for each frame.

Output: $(I^*, \mathbf{u}^*, k^*)$

I^*: image of the tracked face.

$\mathbf{u}^*$: the screen position of I^*.

k^*: current identity of the tracked face.

Model Parameters: $(m, n, L, \boldsymbol{T}, \mathbf{u}^*)$

m: the number of PCA subspaces $L_{k1}, \ldots, L_{km}$ that approximates the submanifolds $C^{k1}, \ldots, C^{km}$ of the appearance manifold M^k of person k.

n: the (common) dimension of the linear subspaces L_{ki}.

L_{ki}: i-th (affine) subspace for person k, represented by a local mean and a set of orthonormal basis vectors.

$\mathbf{T}^k$: a m-by-m probability transition matrix for person k where each entry is an estimated transition probability $p(C^{ki}|C^{kj})$.

$\mathbf{u}^* = (x, y, w, h, \theta)$: the location of the object in the image, represented by a rectangular box in the image centered at (x, y) and of size (w, h) with orientation θ.

Initialization:

The tracker is initialized either manually or by a face detector in the first frame. Let I^* be the initial cropped image from the first frame. Using I^*, the initial identity k^* and the corresponding C^{k^*i} is determined by the minimum L^2 distance between I^* and each pose subspace L_{k^*i}.

Begin

(1) **Sample Windows**: Draw S samples of windows $\{W_1, ..., W_r, ..., W_S\}$ in current image frame specified by $\{u_1, ..., u_r, ..., u_S\}$ at various locations of different orientations and sizes according to a 5-dimensional Gaussian distribution centered at $\mathbf{u}^*$ with diagonal covariance specified by Ω.

(2) **Tracking**: Rectify each window W_r to a 19-by-19 image and rasterize it to form a vector I_r in $\mathbb{R}^{361}$. Compute the L^2 distance between each I_r and the subspace L_{k^*i} associated with C^{k^*i} in the previous frame by evaluating Equation 13. Choose I^* with $\mathbf{u}^*$ that gives the minimal L^2 distance to L_{k^*i} as the tracking output.

(3) **Recognition**: Compute the distance $d(I^*, M_k)$ for each person k using Equations 8 and 9. The identity is computed using Equation 7. Save the corresponding results k^*, C^{k^*i}. Loop back to Step 1 until the last frame.

End

Fig. 6. Summary of the proposed tracking and recognition algorithm.

least 20 s. The resolution of each video sequence is 640 × 480. Every individual is recorded in at least two video sequences. Since we believe that pose variation provides the greatest challenge to recognition, all the video sequences contain significant 2-D (in-plane) and 3-D (out-of-plane) head rotations. In each video, the person rotates and turns his/her head in his/her own preferred order and speed, and typically in about 15 s, the individual is able to provide a wide range of different poses. In addition, some of these sequences contain difficult events which a real-world tracker/recognizer would likely encounter, such as partial occlusion, face partly leaving the field of view, and large scale changes, etc. The data set and all tracking and recognition results are available for download at http://vision.ucsd.edu/kriegman-grp/research/vfr/.

We selected 20 video sequences, one for each individual, for training, and the other 32 sequences are left for testing. The main part of the training procedure is to compute the local linear approximations of each person's appearance manifold as well as the connectivity between these local approximations as detailed in Section 3.2. For this, a simple face tracker (a variant of the EigenTracker of [41,12]) was applied to each training sequence. The tracker returns a cropped face image for each frame, and these cropped images are the training images used to compute the approximation of the appearance manifold for each individual. All of the cropped images produced by the tracker are visually inspected. This manual intervention during the training process is inevitable and necessary because the simple EigenTracker used here is prone to loose the target, and it needs to be re-initialized after each failure. The cropped images are down-sampled to a standard size of 19 × 19 pixels because the tracking windows from different frames are generally of different sizes. Appendix A shows a supplementary experimental result to demonstrate that the cropped images with the standard size of 19 × 19 pixels are effective to perform video-based face recognition. So in these experiments, the appearance manifolds are subsets of $\mathbb{R}^{361}$. Fig. 7 displays some of the cropped and normalized images used as training images.

In our current implementation, the local linear approximation of each appearance manifold contains ten 10-dimensional subspaces. Experiments shown in [11] have demonstrated that this setting effectively captures the structure of the appearance manifold for video-based face recognition. The cropped and normalized images from each individual are grouped into 10 clusters using the K-means clustering algorithm described earlier. A 10-dimensional subspace is computed from the images in each cluster using PCA. As described in the previous section, the connectivity between different subspaces is modeled by a transition matrix M, where each matrix entry $0 \leqslant M_{ij} \leqslant 1$ models the probability that a transition occurs between subspaces indexed by i and j. M_{ij} is computed by counting the number of transitions between subspaces indexed by i and j occurring in a training video sequence (as in Eq. (10)).

4.2. Tracking experiments

Here, we present qualitative and quantitative studies of the effectiveness of our tracking algorithm.

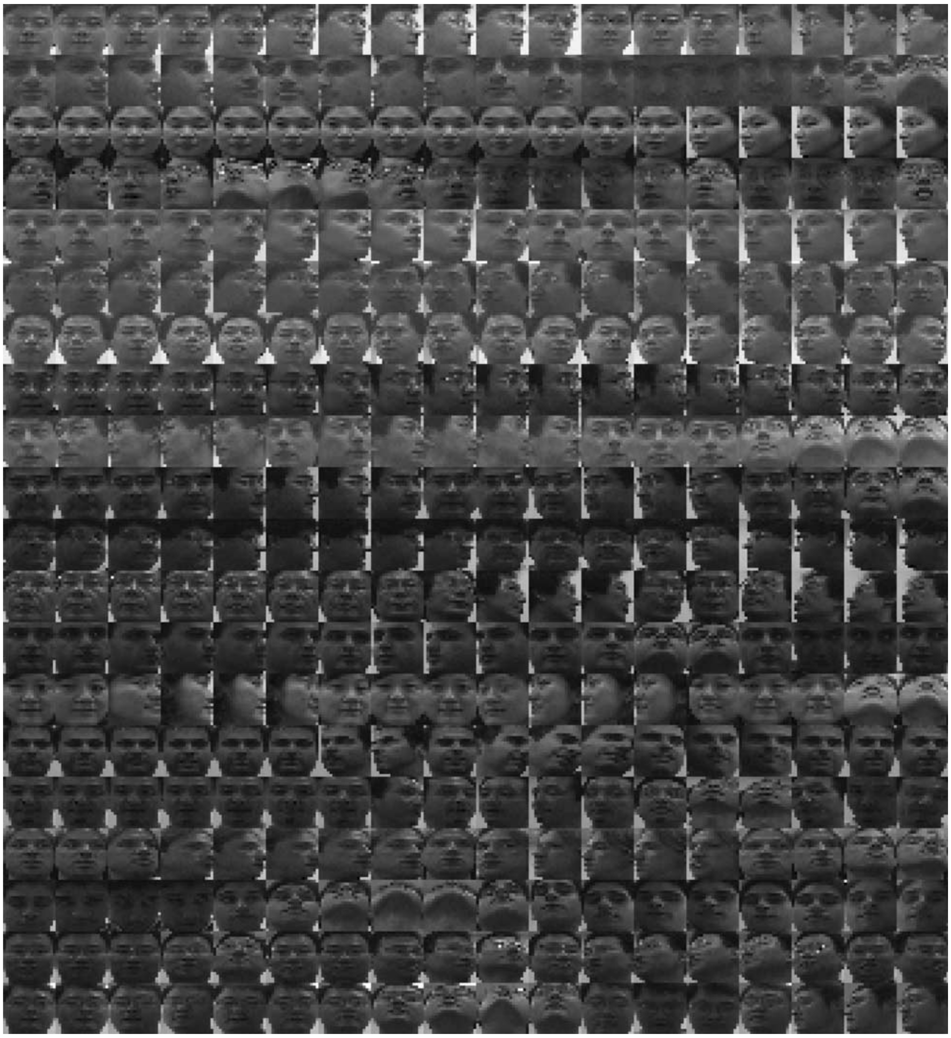
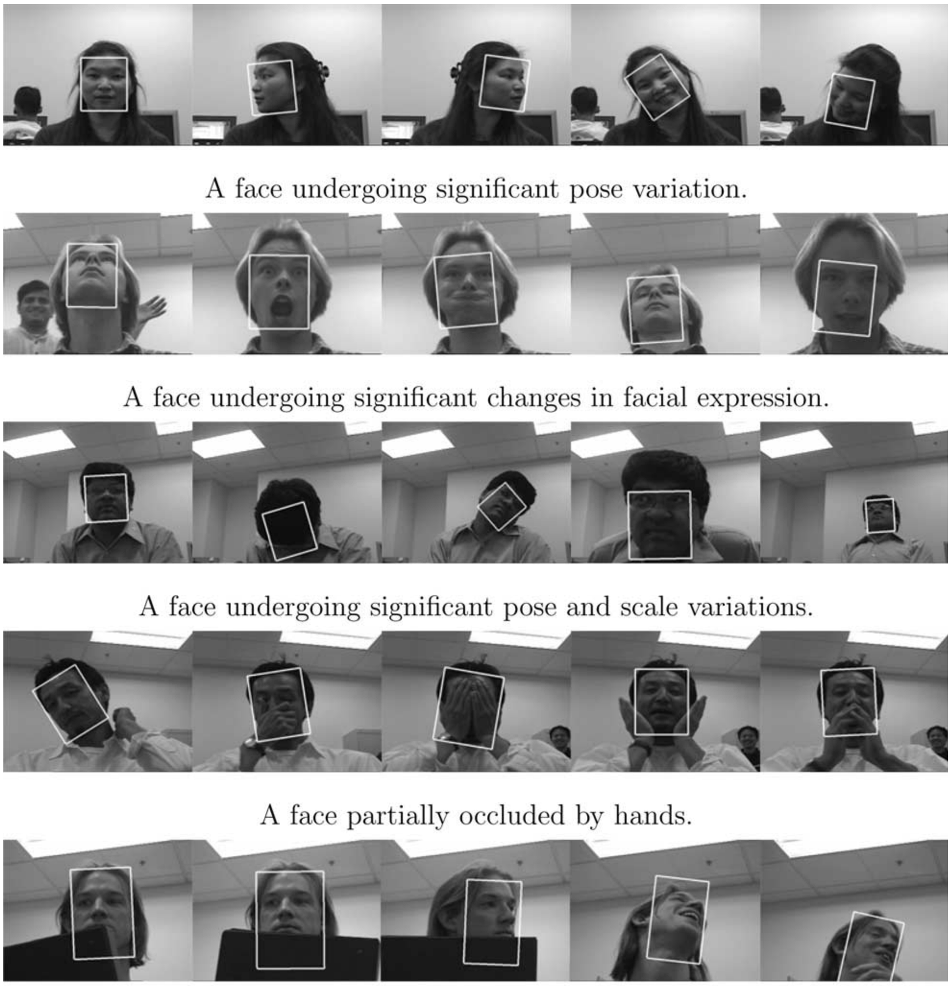


Fig. 7. Samples of the training videos used in the experiments. All sequences contain significant pose variation.

4.2.1. Tracking: qualitative results

Fig. 8 displays the tracking results for five key frames from five different video sequences. The results demonstrate that besides significant pose variation, our tracker is capable of delivering precise tracking results under difficult external conditions including partial occlusion, expression variation as well as large size changes. Note that none of these conditions is present in the training video sequences and hence, they are not modeled by our tracking algorithm. Therefore, the (trained) tracker might not be expected to handle these distractions well. However, the reason why



A face undergoing significant pose variation.

A face undergoing significant changes in facial expression.

A face undergoing significant pose and scale variations.

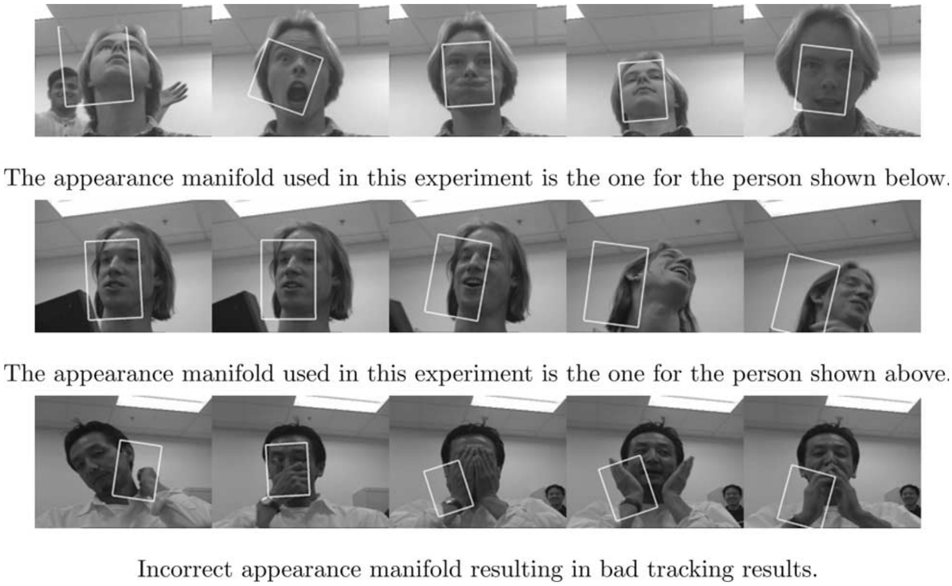
A face partially occluded by hands.

A face partially occluded by a black folder.

Fig. 8. Qualitative tracking results for five different video sequences. Each row displays a set of five key frames from a video sequence.

it quite accurately tracks the intended target is largely because at any instance, the tracker uses an appearance model (linear subspaces) that represents a specific individual identified by the recognizer.

The importance and effect of using the correct appearance model for tracking is illustrated in Fig. 9. In this experiment, the same sequences shown in Fig. 8 are presented to the tracker, but now the incorrect appearance model is used by the tracker. Consequently, the tracking results no longer exhibit the same degree of robustness. Although incorrect appearance models are used, the tracker nevertheless still uses an



The appearance manifold used in this experiment is the one for the person shown below.

The appearance manifold used in this experiment is the one for the person shown above.

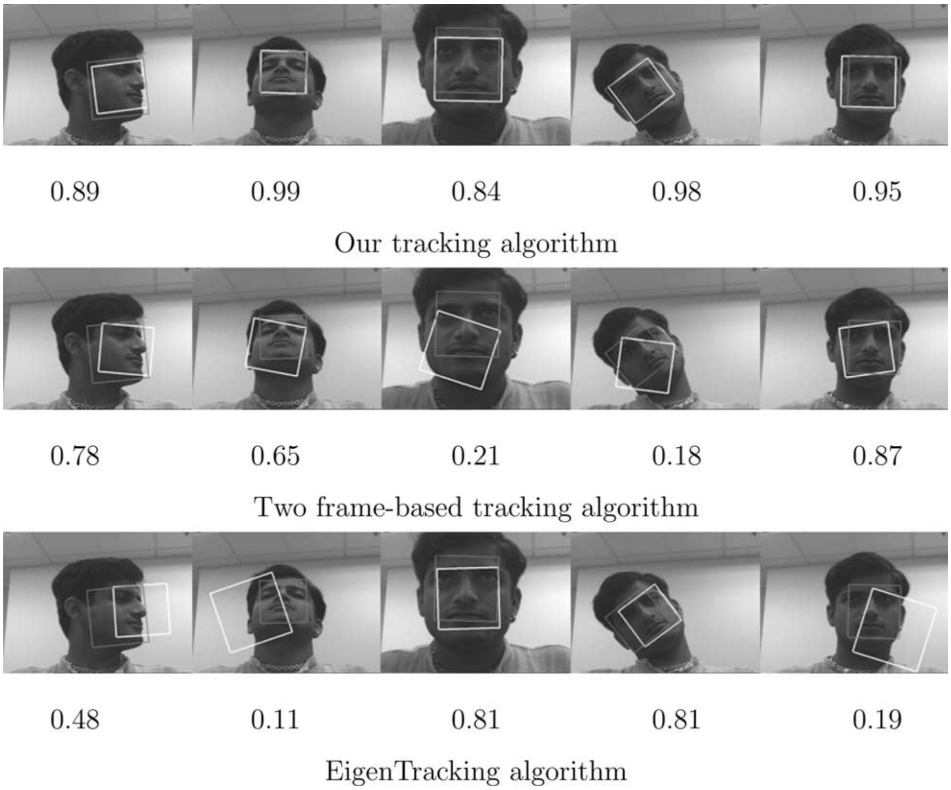
Incorrect appearance manifold resulting in bad tracking results.

Fig. 9. Compared with the tracking results displayed in Fig. 8, the results here are inaccurate. By replacing the correct appearance manifold M_k with a "similar" but different appearance manifold $M_{k'}$, the results are not robust against partial occlusion and expression variation.

appearance model that closely resembles the face in each sequence. These experiments indicate that a detailed and accurate appearance model can provide a certain degree of robustness against partial occlusion and expression changes. One possible explanation is that the more accurate appearance model can provide better matching for the portion of the face that is not occluded or still remains in the same expression. For the inappropriate appearance model, this may not be possible and the tracker's output is usually misaligned. Comparisons with traditional subspace-based trackers [12,6] are also illuminating. These trackers usually employ less specific appearance models,[3] and they require a separate mechanism to handle partial occlusion such as the iterative re-weighted least square method in [6] and nonlinear robust matching in [12]. In contrast, our tracker contains no mechanism specifically for dealing with occlusion.

Tracking results using two other trackers are shown in Fig. 10. The two-frame-based tracker is perhaps the simplest tracker in that the tracker's appearance model consists of only the tracking result from the previous frame. Another tracker used for comparison is a variant of the EigenTracker [12] is a subspace-based algorithm, and in this respect it is technically very similar to ours. The appearance model of the tracker is defined by one single 30-dimensional PCA subspace computed from our training images. However, our implementation of this EigenTracker differs from

[3] In a sense, our appearance model is person-specific while the general subspace-based trackers do not use appearance models as detailed as ours.



Our tracking algorithm

Two frame-based tracking algorithm

EigenTracking algorithm

Fig. 10. Qualitative comparison among our tracker, two-frame-based tracker, and EigenTracker. For each frame, the tracking result is shown with a white rectangle, and the ground truth is represented by a gray thin rectangle. Five key frames, the 22nd, 106th, 127th, 187th, and 287th, of a test sequence totaling 320 frames are displayed. The accuracy value defined by Eq. (14) is shown below each frame.

the original EigenTracker of [12] in two aspects. First, for comparison purpose, the robust matching [12] is omitted since our tracker has no such component. Second and less importantly, the gradient descent in [12] is replaced by the samplings of windows on the screen, exactly as in our tracking algorithm. The major difference between this tracker and ours is that of a single global subspace vs. a collection of local subspaces. As the figure shows, both trackers can still claim that they can stay with the target. However, because the overall aim is to correctly identify the person in the video, misaligned tracking results provided by the two trackers are quite ineffective for any image-based recognition algorithm.

4.2.2. Tracking: quantitative results

To quantitatively evaluate trackers, we need a measure to compare a tracking result (a rectangle) with "ground truth". Let $W^T = (\omega_x^T, \omega_y^T, \omega_w^T, \omega_h^T, \omega_\theta^T)$ and $W^G = (\omega_x^G, \omega_y^G, \omega_w^G, \omega_h^G, \omega_\theta^G)$ denote two rectangular regions in an image, where

(ω_x, ω_y) specifies the center of the rectangle, (ω_w, ω_h) specifies its width and height, and ω_θ denotes the angular orientation. The similarity between these two rectangular regions is defined as

$$S(W^T, W^G) = \exp\left(\sum_{i=1}^{5} \frac{-(\omega_i^T - \omega_i^G)^2}{\sigma_i^2}\right), \tag{14}$$

where the weights, σ_i, are used to control the sensitivity of the similarity measure. The exponential ensures that the largest value is 1, and this occurs only when the two rectangular regions coincide.

With the similarity between two rectangular regions defined by Eq. (14), the accuracy of a tracking result can be defined by computing the similarity between the rectangular region returned by the tracker and a rectangular region deemed as the "ground truth." To obtain the "ground truth," we manually inspect all images in a video sequence and select a rectangular region containing the face. Fig. 10 displays several accuracy values computed for five frames of a video sequences (with 319 frames) using three different trackers, and Fig. 11 plots the accuracy values of the entire sequence. In the plot, the tracker is considered "lost" if the accuracy value is smaller than 0.1. Once the tracker is declared lost, we re-initialize the tracker to the rectangular region represented by the "ground truth" and continue the tracking process. The re-initializations are seen in the plot by the

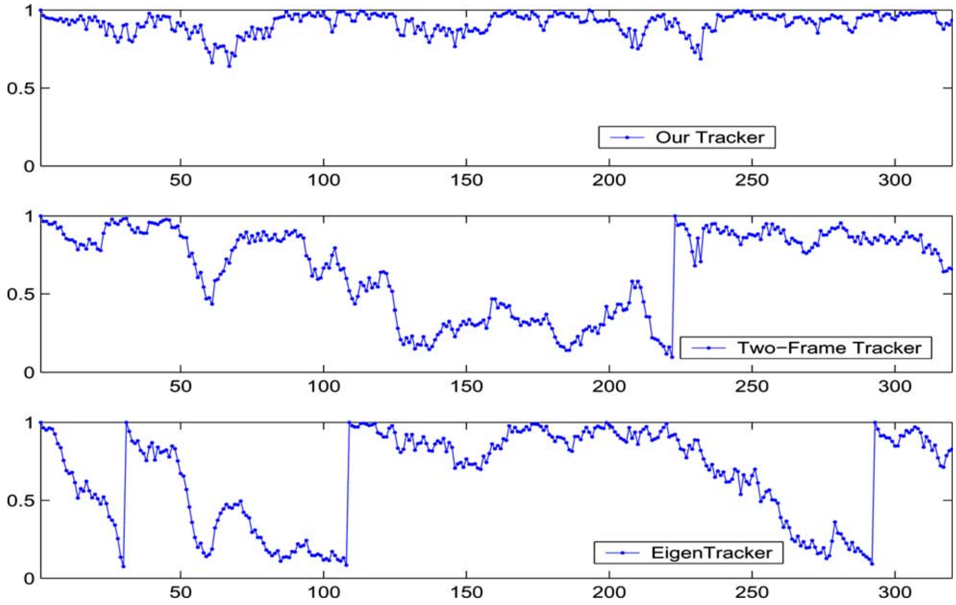


Fig. 11. Quantitative comparisons among our tracker, a two frame-based tracker, and the EigenTracking algorithm for the test sequence shown in Fig. 10. The abscissa represents the frame number, and the ordinate represents the accuracy defined by Eq. (14).

vertical jumps. In these experiments, the values for $\sigma_1, \ldots, \sigma_5$ are set to 8.45, 6.39, 9.64, 10.10, 4.31, respectively. These numbers are determined as follows. We apply our tracker to all the test video sequences. For each i, $1 \leqslant i \leqslant 5$ and each frame, we compute $\omega_i^T - \omega_i^G$, where ω_i^G is the "ground-truth" result and ω_i^T is our trackers' output. Each σ_i is simply the standard deviation of $\omega_i^T - \omega_i^G$ gathered from all the test video sequences.

As can be seen from these plots, our tracker is almost always more accurate than the other two trackers. The two-frame-based tracker was declared lost only once, but in general it was less accurate than the other two competing trackers. The plot for the two-frame-based tracker also indicates the inevitable error accumulated by the tracker; the accuracy degrades continuously and noticeably throughout the sequence. The EigenTracker, on the other hand, is more accurate than the two frame-based tracker on average. The EigenTracker works well when the images lies close to the Eigen-Subspace. However, there are certain portions of the video sequence which are poorly modeled with the single subspace used by the EigenTracker. The result is significant degradation in tracking accuracy for these frames, and almost invariably, it causes the tracker to lose the target.

An alternative criterion to compare the robustness of the trackers is to count the number of times a tracker loses the target. For the results in Fig. 11, we have used a threshold value of 0.1 to define failure of a tracker, and certainly this value can be varied. Fig. 12 shows a plot of the number of tracking failures for the three trackers as the threshold is varied. This demonstrates that our tracker outperforms the other two trackers by a significant margin.

The experimental results reported in this section have provided direct empirical support for our claim that in a face tracking/recognition system, the two processes

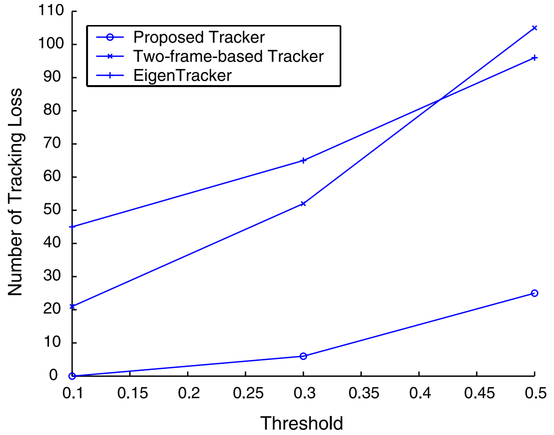


Fig. 12. In the plot, the abscissa denotes the threshold values used to define tracking failures, and the ordinates represents the number of failures. Once the tracker is declared lost (when the accuracy value is smaller than the threshold value), it is re-initialized and continued. A total of 32 videos with 9866 frames were used in this experiment.

should not be completely independent from each other. Instead, each should enhance the performance of the other. In our algorithm, the recognizer provides the person's identity for the tracker, and the tracker selects for each frame the appropriate appearance model to use based on the recognition result. The results reported above strongly indicate that a robust and stable tracker can be obtained based on the simple principle of integrating a tracker and a recognizer.

4.3. Recognition results

In this section, we report on three different sets of experiments to test various aspects of the recognition component of our algorithm. Simple video face recognition systems can be implemented by performing face recognition at each frame independently using existing techniques such as Eigenfaces [10] and Fisherfaces [42]. Our algorithm differs from these frame-based algorithms through the use of a probabilistic model that integrates recognition results across different frames. The first set of experiments demonstrates that these frame-based recognition methods do not have comparable recognition performance to ours. With the importance of using temporal information established, the second set of experiments shows that our probabilistic model offers considerably better robustness and stability than the usual majority voting method, one of the most popular and simplest methods for integrating recognition results across time. While the first two sets of experiments demonstrate that our algorithm provides robust and stable recognition results for video sequences containing a single individual, the third set of experiments shows that our algorithm can also swiftly and correctly detect a change in identity.

4.3.1. Comparisons with frame-based algorithms

Table 1 shows the results of comparing our algorithm with four frame-based recognition algorithms. The error rates are computed by taking the ratio of the number of correctly recognized frames and the total number of frames used in the experiment.[4] All algorithms listed in Table 1 use the same collection of training images, and excluding our algorithm, they simply perform face recognition for each frame using their respective methods. For our algorithm, each individual is represented by ten 3-D local PCA subspaces, and the transition probabilities. The Ensemble of LPCA (eLPCA for short) shares the same collection of local linear subspaces as our method, but without using the transition probabilities. eLPCA computes the distance between each test image and all local PCA subspaces, and it returns the identity associated with the subspace that gives the minimal distance to the test image. This method can be considered as using a set of local linear models to approximate a global appearance manifold without defining any connectivity between these local models.

[4] A total of 32 video sequences of 20 individuals with 9866 frames were used in all the recognition experiments. Eleven video sequences with 3186 frames contain significant partial occlusions. The other 21 video sequences with 6680 frames contain only pose and expression variations.

Table 1
Recognition accuracy comparing frame-based methods and the proposed method

Method	Accuracy (%)	
	Videos w/o occlusion	Videos with occlusion
Comparison of recognition methods		
Proposed method	98.8	97.8
Ensemble of LPCA	76.9	70.3
Eigenfaces	69.3	53.7
Fisherfaces	74.5	65.4
Nearest Neighbor	81.6	76.3

The next three methods are all standard image-based face recognition algorithms. The linear projection space used by Fisherfaces [42] is set to 19 (i.e., the number of classes minus 1). As for the Eigenfaces method [10], a global PCA subspace of dimension 30 is used to linearly reduce all training data. In both Fisherfaces and Eigenfaces methods, all training images are projected to the respective subspaces and the usual *K*-means clustering algorithm is applied to the projected data to yield 40 clusters for each individual.[5] For each test image, both algorithms compute the distances between the projected test image and these cluster centers, and the algorithms return the identity associated with the cluster centers that gives the minimal distance. The Nearest Neighbor method (NN) does not require any projection, and the training images are directly clustered in the image space to yield 40 clusters for each individual. The rest of the algorithm is the same as in Eigenfaces.

The results in Table 1 demonstrate that our algorithm offers better recognition performance. Furthermore, the results with the occlusion sequences illustrate the robustness and stability of our algorithm. While our algorithm barely acknowledges the challenges posed by the occlusion sequences, the recognition performance of all other algorithms has degraded considerably. Though it may not seem to be fair to compare with frame-based recognition algorithms, these baseline experiments suggest that frame-based methods may not work well in an unconstrained environment where there are large pose changes. There are essentially two important problems as we mentioned earlier, the nonlinear nature of the appearance manifold due to significant pose variations and our inability to densely sample images from it. The comparison between eLPCA and the two classical linear methods (Eigenfaces and Fisherfaces) illustrate the first point. Because eLPCA uses local linear subspaces to approximate the nonlinear appearance manifold, it is expected to provided a more accurate approximation, especially when compared with Eigenfaces which uses only one subspace for modeling pose variation. While eLPCA uses local linear approximations, the Nearest Neighbor directly samples points from the appearance manifold. We have used forty images for

[5] In our algorithm, each individual is represented by ten affine subspaces of dimension three. Each subspace is represented by its center and three orthogonal vectors. This is equivalent to four images (vectors), and hence each individual is represented by 40 images (vectors) in our algorithm.

each individual in the Nearest Neighbor method. This is by no mean a dense sampling of the appearance manifold, and therefore, there are ''gaps'' in which in which the person's appearance cannot be adequately modeled. In a frame-based algorithm, these ''gaps'' are problematic and they usually cause incorrect recognition. Denser sampling would likely produce better recognition results, but at the expense of requiring longer computation. The experimental results show that our algorithm correctly fills in these gaps most of the time.

The conclusion we draw from the experiments is quite different from that reported by the Face Recognition Vendor Test 2002 (FRVT 2002) [43,44], which claimed that temporal information (i.e., videos) does not enhance nor improve face recognition performance. In particular, there are comparisons in FRVT 2002 that show no significant difference between video and still-image face recognitions. Since no detail of the video face recognition algorithms used by the vendors are available to us, we do not know if algorithms similar to ours have been tested using FRVT 2002 video data. However, we notice that most of the images used in FRVT 2002 are well-cropped frontal face images, and large pose variation and partial occlusion rarely appear in test images. In our framework, the frontal pose is modeled by just one subspace. Applying our algorithm to FRVT 2002 image data would probably result in *no* transition between subspaces, and hence, no temporal information being used. This indicates the possibility that the conclusion concerning video face recognition presented in [43,44] may be more data-specific instead of a more general observation.

4.3.2. The effect of transition matrix $P(C^{ki}|C^{kj})$

In this set of experiments, we demonstrate that the transition matrix, $p(C^{ki}|C^{kj})$, in our algorithm does capture the image dynamics sufficiently to improve recognition rates. Certainly, other temporal strategies for integrating recognition results across different frames are possible. Temporal voting (or majority voting) is the most well-known (e.g. [20,45,22]), and we augmented the eLPCA method described earlier with a temporal voting scheme (using voting windows of size 30). The additional recognition algorithm (Temporal Voting in Table 2) is similar to our algorithm except that our algorithm has a more sophisticated probabilistic model for integrating temporal information. For the second method (Uniform Transition in Table 2), we ran our algorithm with all the entries in the transition matrix set to a default constant. The main difference between this algorithm and ours is that although transitions between different subspaces are summed in both algorithms, this algorithm counts all

Table 2
Recognition results using various temporal strategies

Temporal strategy	Accuracy (%)	
	Videos w/o occlusion	Videos with occlusion
Comparison of temporal strategies		
Proposed method	98.8	97.8
Temporal Voting	84.2	74.4
Uniform Transition	80.1	70.2

transitions with equal weight (same $p(C^{ki}|C^{kj})$, see Eq. (9)) while ours associates each transition with different weights (likelihood $p(C^{ki}|C^{kj})$) learned from the training video sequences.

The experimental results demonstrate that utilizing the transition probabilities does improve recognition performance. Again, the results for occlusion sequences are quite striking in that the two methods suffer from substantial performance degradations at the same time our algorithm holds its own ground quite well. The comparison between Temporal Voting and eLPCA is also quite illuminating. While as expected the Temporal Voting method (being based on eLPCA) outperforms eLPCA, it suffers more serious performance degradation than eLPCA for the challenging occlusion sequences.

4.3.3. Abrupt changes in identity

The previous two experiments have demonstrated the stability and robustness of our recognition algorithm. In particular, after converging quickly to the correct identity, the algorithm almost always returns the same identity in the subsequent frames. Because every video sequence used in the experiments contains only one individual, it is possible that our algorithm, after the initial convergence, becomes stationary and gets stuck with the same identity. Other works in video face recognition (e.g. [2]) have also reported similar fast initial convergence of the recognition result. One possible way to investigate this issue is to re-initiate the recognizer every time after the convergence has been reached and stabilized such as in [2]. Since there is no such mechanism in our algorithm, we need to demonstrate that the stability of our algorithm is not the consequence of the algorithm's fixation on one particular individual. To this aim, we show that our algorithm can swiftly and correctly detect changes in identity when the situation demands.

For this experiment, we created 500 image sequences by concatenating two consecutive segments from two video sequences of different individuals, who were selected at random. Since these unnatural image sequences are usually difficult for the tracker to render correct results, instead we concatenate sequences of *cropped images* that we gathered in the tracking experiment. The first part of each sequence contains 100 consecutive frames from one individual and the second part contains 50 frames from another individual. The objective of the experiment is to allow our recognizer to stabilize its recognition result during the first 100 frames and then to examine its response to a sudden and abrupt change in identity. Our algorithm is tested on all 500 image sequences and for each sequence, we record the number of frames needed for the recognizer to correctly identify the individual in the 101th frame. Fig. 13 displays a histogram of the experiment results.

For the 500 sequences, our algorithm requires, on average, 3.5 frames to recover the correct identity, with standard deviation of 4.8 frames and median of 2 frames. Surprisingly, the recognizer can immediately detect the change in about 20% of the sequences, and for more than 90% of the sequences, it takes fewer than ten frames for our recognizer to recover the correct identity. Depending on the video's frame rate, this roughly corresponds to one third of a second, which is acceptable for many applications. For the Temporal Voting method (with window size 30), it takes, on

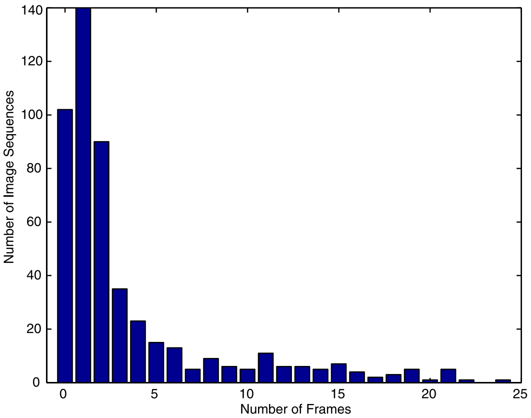


Fig. 13. A histogram of the experiment results. The abscissa denotes the number of frames needed for our algorithm to switch to the correct identity. The ordinate axis denotes the number of image sequences. 500 image sequences were used in the experiment.

average, 15 frames for the algorithm to correctly detect the change. Both our method and the Temporal Voting method integrate recognition results over time; therefore, lags in their correct responses to identity change are expected. On the other hand, the single-frame based algorithm can usually detect the change immediately because previous recognition results have no influence on the present recognition process. In this experiment, the eLPCA method almost always correctly identifies the change after only one frame. However, as we mentioned before, eLPCA method does not give comparable recognition performance to our method, and for the 500 image sequences, the overall recognition rate for our method and for eLPCA are 93.41 and 78.92%, respectively (compare with Table 1). Overall, our algorithm achieves superior recognition performance with a small lag in response time in the event of an identity change.

5. Conclusion and future work

We have presented a novel framework for face recognition and tracking in video sequences that tightly couples tracking and recognition components. In this new framework, both the tracking and recognition components share the same appearance model to minimize the misalignment between the tracker's output and the recognizer's input. The appearance of each face is modeled by a collection of linear subspaces in the image space. Specifically, each PCA subspace approximates a collection of training images with similar appearances. Conceptually, the collection of subspaces constitutes a piecewise linear approximation of the object's appearance manifold [9]. The connectivity of the appearance manifold is represented in our framework by a matrix of transition probabilities between pairs of subspaces. The transition matrix is learned directly from the training video sequences by observing the actual transitions between different pose states. We have also proposed a Bayesian inference framework to integrate recognition results computed independently at each frame to yield a final robust recognition decision. The experimental results have demonstrated that our novel framework is capable of providing robust and stable results for video face recognition.

We have collected video sequences that are challenging for both tracking and recognition. The video sequences used in this paper all contain significant 2-D and 3-D rotations as well as many other challenging real-world "disturbances," such as partial occlusion and expression variation. However, illumination variation is an important class of image variation that is not modeled by our algorithm. Though our algorithm handles large pose and expression variations well, it is nevertheless sensitive to large illumination changes. Presently, histogram equalization is used to deal with serious illumination changes. However, a more attractive alternative may be to incorporate the idea of the illumination cone [46] directly into our framework in order to model image variation under illumination changes. Numerous subspace-based face recognition methods [47,8,48] have demonstrated their robustness against significant illumination variation. Therefore, an interesting direction for future research is to formulate a new and more inclusive subspace-based framework under which both pose and illumination variations can be handled.

Acknowledgments

Support for this work was provided by Honda Research Institute, and the National Science Foundation CCR 00-86094 and IIS 00-85980. The video sequences used in the experiments as well as a preliminary version of the face recognition system were recorded and implemented during the summer of 2002 at Honda Research Institute.

Appendix A. Recognition performance vs. resolution of cropped images

In this appendix, we present a quantitative study of the impact on face recognition of down-sampling the tracking window to different resolutions. In this experiment, we down-sampled the tracking windows in the training videos to the given resolution, and constructed the proposed representation using clustering and PCA. Tracking and recognition then proceed as in Section 3.5. The resolutions vary by a factor of 1.3. Fig. 14 displays the recognition result of our proposed algorithm and the standard Eigenfaces method. Except for the resolution of the down-sampled cropped images, all of the parameter settings are the same as presented in Section 4.

The results show that the recognition rate is flat for both recognition algorithms when the down-sampled image size ranges from 19-by-19 to 42-by-42, and the performance drops dramatically when the image size becomes smaller than 16-by-16 for our method. For Eigenfaces method, the performance is again flat but for all reso-

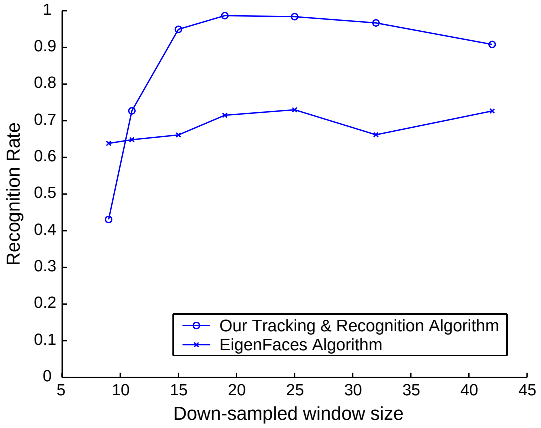


Fig. 14. In the plot, the abscissa denotes the down-sampled size of the cropped face windows, and the ordinates represents the the recognition rate. A total of 20 videos with 6076 frames were used in this experiment.

lutions. Therefore, for the computational efficiency, we choose 19-by-19 as the down-sampling size of the cropped face images for all the experiments reported in the Section 4.

References

[1] G.J. Edwards, C.J. Taylor, T.F. Cootes, Improving identification performance by integrating evidence from sequence, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1999, pp. 486–491.

[2] V. Krueger, S. Zhou, Exemplar-based face recognition from video, In: Proc. European Conf. on Computer Vision, vol. 4, 2002, pp. 732–746.

[3] Y. Li, S. Gong, H. Liddell, Video-based online face recognition using identity surfaces, in: Proc. Internat. Conf. on Computer Vision, vol. 1, 2001, pp. 554–559.

[4] X. Liu, T. Chen, Video-based face recognition using adaptive hidden markov models, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 26–33.

[5] S. Birchfield, Elliptical head tracking using intensity gradients and color histograms, In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1998, pp. 232–237.

[6] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, in: IEEE Trans. Pattern Analysis and Machine Intelligence, 1998, pp. 1025–1039.

[7] R. Brunelli, T. Poggio, Face recognition: features versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15 (10) (1993) 1042–1052.

[8] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.

[9] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, Internat. J. Comput. Vision 14 (1995) 5–24.

[10] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 1994, pp. 84–91.

[11] K.-C. Lee, J. Ho, M.-H. Yang, D. Kriegman, Video-based face recognition using probabilistic appearance manifolds, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2003, pp. 313–320.

[12] M.J. Black, A.D. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, Internat. J. Comput. Vision 26 (1) (1998) 63–84.

[13] A. Jepson, D. Fleet, T. El-Maraghi, Robust online appearance models for visual tracking, in: IEEE Trans. Pattern Analysis and Machine Intelligence, 2003, pp. 1296–1311.

[14] Y. Chen, Y. Rui, T. Huang, JPDAF based HMM for real-time contour tracking, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2001, pp. 543–550.

[15] Y. Wu, T.S. Huang, A co-inference approach to robust visual tracking, in: Proc. Internat. Conf. on Computer Vision, vol. 2, 2001, pp. 26–33.

[16] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 142–149.

[17] R. Chellappa, C.L. Wilson, S. Sirohey, Human and machine recognition of faces: a survey, Proc. IEEE 83 (5) (1995) 705–740.

[18] A. Samal, P.A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: a survey, Pattern Recogn. 25 (1) (1992) 65–77.

[19] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surveys 35 (4) (2003) 399–458.

[20] A.J. Howell, H. Buxton, Towards unconstrained face recognition from image sequences, in: Proc. IEEE Internat. Conf. on Automatic Face and Gesture Recognition, 1996, pp. 224–229.

[21] G. Shakhnarovich, J.W. Fisher, T. Darrell, Face recognition from long-term observations, in: Proc. European Conf. on Computer Vision, vol. 3, 2002, pp. 851–865.

[22] H. Wechsler, V. Kakkad, J. Huang, S. Gutta, and V. Chen, Automatic video-based person authentication using the RBF network, in: Proc. Internat. Conf. on Audio and Video-Based Biometric Person Authentication, 1997, pp. 177–183.

[23] S. Zhou, R. Chellappa, Probabilistic human recognition from video, in: Proc. European Conf. on Computer Vision, vol. 3, 2002, pp. 681–697.

[24] K. Toyama, A. Blake, Probabilistic tracking in a metric space, in: Proc. Internat. Conf. on Computer Vision, vol. 2, 2001, pp. 50–59.

[25] Y. Li, T. Wang, H. Shum, Motion textures: a two-level statistical model for character motion synthesis, in: Proc. SIGGRAPH, 2002, pp. 465–472.

[26] C.M. Bishop, J.M. Winn, Non-linear Bayesian image modelling, in: Proc. European Conf. on Computer Vision, vol. 1, 2000, pp. 3–17.

[27] C. Bregler, S. Omohundro, Surface learning with applications to lipreading, in: Advances in Neural Information Processing Systems, 1994, pp. 43–50.

[28] M. Isard, A. Blake, A mixed-state Condensation tracker with automatic model-switching, in: Proc. Internat. Conf. on Computer Vision, 1998, pp. 107–112.

[29] B. North, A. Blake, M. Isard, J. Rittscher, Learning and classification of complex dynamics, IEEE Trans. Pattern Anal. Mach. Intell. 22 (9) (2000) 1016–1034.

[30] V. Pavlović, J.M. Rehg, T.J. Cham, K.P. Murphy, A dynamic Bayesian network approach to figure tracking using learned dynamic models, in: Proc. Internat. Conf. on Computer Vision, 1999, pp. 94–101.

[31] A. Schödl, R. Szeliski, D.H. Salesin, I. Essa, Video textures, in: Proc. SIGGRAPH, 2000, pp. 489–498.

[32] T. Cootes, C.J. Taylor, D. Cooper, J. Graham, Active shape models—their training and application, Comput. Vision Image Understand. 61 (1995) 38–59.

[33] X. Hou, S. Li, H. Zhang, Q. Cheng, Direct appearance models, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 828–833.

[34] T. Jebara, K. Russell, A. Pentland, Mixtures of Eigen Features for real-time structure from texture, in: Proc. Internat. Conf. on Computer Vision, 1998, pp. 128–135.

[35] D. DeCarlo, D. Metaxas, M. Stone, An anthropometric face model using variational techniques, in: Proc. SIGGRAPH, 1998, pp. 67–74.

[36] G.J. Edwards, C.J. Taylor, T.F. Cootes, Interpreting face images using active appearance models, in: Proc. IEEE Internat. Conf. on Automatic Face and Gesture Recognition, 1998, pp. 300–305.

[37] W.Y. Zhao, R. Chellappa, Symmetric shape-from-shading using self-ratio immage, Internat. J. Comput. Vision 45 (1) (2001) 55–75.

[38] S. Romdhani, V. Blanz, T. Vetter, Face identification by fitting 3D morphable model using linear shape and texture error functions, in: Proc. European Conf. on Computer Vision, vol. 3, 2002, pp. 3–19.

[39] D. Hochbaum, D. Shmoys, A best possible heuristic for the k-center problem, Math. Operations Res. 10 (1985) 180–184.

[40] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, in: Internat. J. Computer Vision, 1998.

[41] M. Black, A. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, in: Proc. European Conf. on Computer Vision, 1996, pp. 329–342.

[42] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997) 711–720.

[43] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, J. Bone, FRVT 2002: overview and summary, in: 2002 Face Recognition Vendor Test, 2003. Available from: <http://www.frvt.org/FRVT2002/documents.htm>.

[44] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, J. Bone, FRVT 2002: evaluation report, in: 2002 Face Recognition Vendor Test, 2003. Available from: <http://www.frvt.org/FRVT2002/documents.htm>.

[45] Y. Li, S. Gong, H. Liddell, Constructing facial identity surface in a nonlinear discriminating space, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 258–263.

[46] P. Belhumeur, D. Kriegman, What is the set of images of an object under all possible lighting conditions? Internat. J. Comput. Vision 28 (1998) 245–260.

[47] R. Basri, D. Jacobs, Lambertian reflectance and linear subspaces, in: Proc. Internat. Conf. on Computer Vision, vol. 2, 2001, pp. 383–390.

[48] K.-C. Lee, J. Ho, D. Kriegman, Nine points of light: acquiring subspaces for face recognition under variable lighting, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2001, pp. 519–526.